MSBD 5002 Homework 2

Pranav A, 20478966

November 13, 2017

1 Nearest Neighbors

According to the question, we get the following distance matrix:

Testset item	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
15	4	3	3	3	3	5	4	3	5	3	3	3	4	2
16	4	3	3	3	1	3	3	3	3	3	2	3	5	3
17	3	4	5	2	5	4	4	5	4	4	3	2	4	3
18	4	3	3	4	3	3	3	3	3	3	5	2	4	4
19	2	4	4	4	3	2	3	4	3	4	3	1	4	2
20	2	3	2	4	3	2	3	2	1	4	4	4	3	5

Table 1: Distance Matrix

For $k = 1$ and $k = 4$, we get the following result matrix (tie is breaked by choosing the lowest index):

Testset Item	**Actual Score**	**Predicted Score** ($k = 1$)	**Predicted Score** ($k = 4$)
15	High	Medium	Medium
16	Medium	Low	Medium
17	High	Low	Low
18	Medium	Low	High
19	Low	Low	High
20	Medium	Medium	Medium

Table 2: Testset prediction

Here, I get $\frac{2}{3}$ as the misclassification error for $k = 1$ and $\frac{5}{6}$ as the misclassification error for $k = 4$.

2 Decision Trees

C4.5 has less steps to calculate than the ID3 algorithm. C4.5 uses information gain ratio to split the trees. C4.5 uses bottom-up approach but ID3 uses top-down approach. Thus C4.5 could be useful in pruning. This also reduces the criteria of overfitting too. More importantly, C4.5 handles both discrete and continuous features.

3 Naive Bayes Classifier

3.1 Calculations

The probabilities of the target classes are given by:

$$P(Low) = \frac{1}{4}$$
$$P(Medium) = \frac{9}{20}$$
$$P(High) = \frac{3}{10}$$

The conditional probability of each attribute is given by,
For the class Low:

$$P(Language = NonEnglish|Low) = \frac{2}{5}$$
$$P(Instructor = David|Low) = \frac{1}{5}$$
$$P(Course = Database|Low) = \frac{3}{5}$$
$$P(Semester = 2|Low) = \frac{4}{5}$$

For the class Medium:

$$P(Language = NonEnglish|Medium) = \frac{8}{9}$$
$$P(Instructor = David|Medium) = \frac{1}{3}$$
$$P(Course = Database|Medium) = 0$$
$$P(Semester = 2|Medium) = \frac{2}{3}$$

For the class High:

$$P(Language = NonEnglish|High) = \frac{1}{2}$$
$$P(Instructor = David|High) = \frac{1}{2}$$
$$P(Course = Database|High) = \frac{1}{3}$$
$$P(Semester = 2|High) = \frac{1}{2}$$

We know that the Naive Bayes formula that:

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

Where C denotes the classes and the x denotes the given attribute.
Calculating the probabilities of each class now:

$$\begin{aligned}
\hat{y}_{Low} &= \; p(Low) \prod_{i=1}^{n} p(x_i \mid Low) \\
&= \frac{6}{625} \\
\hat{y}_{Medium} &= \; p(Medium) \prod_{i=1}^{n} p(x_i \mid Medium) \\
&= 0 \\
\hat{y}_{High} &= \; p(High) \prod_{i=1}^{n} p(x_i \mid High) \\
&= \frac{1}{80}
\end{aligned}$$

Clearly, the maximum value here is of the **High** class. Hence the predicted score of TA is **High**.

There is no need of smoothing here as conditional probabilities of Medium won't change that dramatically and High class would still be predicted.

3.2 Why it is called Naive

:

This classifier assumes the conditional independence between the classes. Also, the classifier does not interpolate well within the classes. In reality, there would be some dependencies between the classes. Hence, such assumption is too basic and naive. But it kind of sets some baselines and provides basic overview, despite being called as "naive".

4 Bayesian Networks

4.1 Inequality Conditions

$$
\begin{aligned}
& P(E=y|A=y) > P(E=y|A=y,B=y) \\
\implies & \frac{P(A=y|E=y)\times P(E=y)}{P(A=y)} > \frac{P(A=y|E=y,B=y)\times P(E=y,B=y)}{P(A=y|B=y)\times P(B=y)} \\
\implies & \frac{(\alpha_1+\alpha_3)\times 0.1}{\alpha_1+\alpha_2+\alpha_3+\alpha_4} > \frac{\alpha_1\times 0.2\times 0.1}{(\alpha_1+\alpha_2)\times 0.2} \\
\implies & \frac{\alpha_1+\alpha_3}{\alpha_1+\alpha_2+\alpha_3+\alpha_4} > \frac{\alpha_1}{\alpha_1+\alpha_2} \\
\implies & (\alpha_1+\alpha_3)\times(\alpha_1+\alpha_2) > \alpha_1\times(\alpha_1+\alpha_2+\alpha_3+\alpha_4) \\
\implies & \alpha_2\times\alpha_3 > \alpha_1\times\alpha_4
\end{aligned}
$$

4.2 Values

Suppose $\alpha_1=\alpha_4=\frac{1}{1000}$ and $\alpha_2=\alpha_3=\frac{1}{10}$. Plugging these values, we get:
Distribution of the $P(A|B,E)$:

$$
\begin{aligned}
P(A=y|B=y,E=y) &= \frac{1}{1000} \\
P(A=y|B=y,E=n) &= \frac{1}{10} \\
P(A=y|B=n,E=y) &= \frac{1}{1000} \\
P(A=y|B=n,E=n) &= \frac{1}{10} \\
P(A=n|B=y,E=y) &= \frac{999}{1000} \\
P(A=n|B=y,E=n) &= \frac{9}{10} \\
P(A=n|B=n,E=y) &= \frac{999}{1000} \\
P(A=n|B=n,E=n) &= \frac{9}{10}
\end{aligned}
$$

Left hand side:

$$P(E=y|A=y)=\frac{101}{2002}$$

Right hand side:

$$P(E=y|A=y,B=y)=\frac{1}{10010}$$

Thus, left hand side is larger than the right hand side.

5 Adaboost

5.1 Implementation

Here I will reject the h_3 classifier, as it is only outputting the positive labels. Thus this classifier would not be useful in the ensemble.

The classifiers chosen are h_1 and h_2. Let the weights assigned to h_1 and h_2 be w, where $w = [0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125]$. Here the error function is defined by $e_h = \sum I(h(x) \neq y)$, where I is the indicator function. Basically, error function counts number of mismatches.

Iteration 1:

Here we will find the weighted sum errors of each classifiers.

$$\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq y_i}}^{n} w_{i,t}$$

$$\epsilon_1 = \sum_{\substack{i=1 \\ h_1(x_i) \neq y_i}}^{n} w_{i,1} = 0.375$$

$$\epsilon_2 = \sum_{\substack{i=1 \\ h_2(x_i) \neq y_i}}^{n} w_{i,2} = 0.5$$

Here we will move forward with stronger classifier, ϵ_1. Next step is to calculate α.

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

$$\alpha_1 = \frac{1}{2} \ln \left(\frac{1 - \epsilon_1}{\epsilon_1} \right) = 0.2554$$

Update weights of w, we get after normalizing with sum of weights:
$w = [0.1, 0.167, 0.1, 0.1, 0.167, 0.167, 0.1, 0.1]$

Iteration 2

Here we will find the weighted sum errors of each classifiers.

$$\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq y_i}}^{n} w_{i,t}$$

$$\epsilon_1 = \sum_{\substack{i=1 \\ h_1(x_i) \neq y_i}}^{n} w_{i,1} = 0.5$$

$$\epsilon_2 = \sum_{\substack{i=1 \\ h_2(x_i) \neq y_i}}^{n} w_{i,2} = 0.533$$

Here $\epsilon_1 \geq 0.5$ and $\epsilon_2 \geq 0.5$. Thus, we will stop the Adaboost here as it cannot proceed further.

Hence the final classifier is

$$F(x) = \text{sign}\left(\sum_t \alpha_t h_t(x)\right)$$

5.2 Response of Strong Classifier

For the final ensemble, we will plug the values, and get the following output:

$$\begin{aligned} F(x) &= \text{sign}\left(\sum_t \alpha_t h_t(x)\right) \\ &= \text{sign}\left(\alpha_1 h_1(x)\right) \\ &= \text{sign}\left(0.2554 \cdot [-1,-1,1,1,1,1,1,-1]\right) \\ &= [-1,-1,1,1,1,1,1,-1] \end{aligned}$$

5.3 Error of Strong Classifier

The classifier makes 3 mismatches from 8, hence the error rate is 37.5%. Here, h_2 did not help with the Adaboost as it was not better than a random classifier. Hence the resulting classifier only comprises of h_1. Again, adaboost is the poor choice for this ensemble. I would have gone for bagging or linear combination of classifiers instead.

Notes

1. This given solution of the assignment follows the HKUST honour code. Although assignment has been discussed with other peers, the solutions are my own.
2. Kindly give constructive feedback for my incorrect or unclear answers.